

# TP3 (version JavaScript)

---

INF8808: Visualisation de données

Département de génie informatique et génie logiciel



**Auteure:** Olivia Gélinas

**Chargé de lab:** Hellen

## Objectifs

---

L'objectif de ce travail pratique est de créer une carte de chaleur interactive en utilisant des données ouvertes en format CSV.

Avant de commencer, nous vous recommandons d'avoir effectué les lectures suivantes et d'avoir effectué les exercices suivants:

**Lectures:** Chapitres 9 (p.149 - p.158), 10 et 11 du livre de Scott Murray

- 
- Exercices:**
- Chapitre 6 - 1, 2, 3
  - Chapitre 8 - 1, 2

## Introduction

---

Une carte de chaleur (*heatmap*) est un type de visualisation de données qui affiche les données sous forme de couleurs représentées selon deux dimensions. La teinte ou intensité de chaque couleur représente la l'intensité de l'observation pour ses valeurs correspondantes en 2 dimensions.

Dans ce travail pratique, vous implémenterez une carte de chaleur interactive en utilisant des données représentant les plantations d'arbres au fil du temps dans les quartiers de Montréal. Les données ont été extraites du portail de données ouvertes de la Ville de Montréal [1]. L'ensemble de données a ensuite été modifié en une version à utiliser dans ce travail pratique. Il contient des données représentant la date et le lieu où les arbres ont été plantés à Montréal.

## Description

---

Dans ce travail pratique, vous devrez compléter le code JavaScript à l'aide de D3 afin d'afficher une carte de chaleur représentant le nombre d'arbres plantés dans chaque quartier de Montréal chaque année entre 2010 et 2020. Pour rendre le graphique interactif, vous implémenterez également le code pour afficher la valeur d'une cellule lorsqu'elle est survolée par la souris. De plus, il y aura une légende indiquant la couleur correspondant à la quantité d'arbres plantés.

Les sous-sections suivantes présentent les différentes parties que vous aurez à compléter pour ce travail pratique. Pendant que vous codez, nous vous recommandons de compléter le traitement des données d'abord, suivi de la mise en œuvre du graphique à barres lui-même. Les deux parties suivantes, la légende et le survol, sont indépendantes.

## Structure des fichiers

Pour réaliser ce travail, vous devrez remplir les différentes sections **TODO** dans les fichiers de l'archive fournie pour le travail pratique. Les commentaires dans le code expliquent plus en détail les étapes à suivre.

Dans ce travail pratique, nous vous fournissons une archive contenant 7 fichiers JavaScript utilisés pour accomplir la visualisation souhaitée:

- **index.js**: Ce fichier représente le point d'entrée du code et orchestre les différentes étapes nécessaires à la réalisation de la visualisation. Il n'a pas besoin d'être modifié.
- **scripts/helper.js**: Ce fichier contient quelques fonctions de base nécessaires pour afficher la visualisation. Il n'a pas besoin d'être modifié.
- **scripts/hover.js**
- **scripts/legend.js**
- **scripts/preprocess.js**
- **scripts/util.js**: Ce fichier contient une petite fonction utilitaire qui est passée en paramètre aux fonctions où elle pourrait être utile. Ce fichier n'a pas besoin d'être modifié.
- **scripts/viz.js**

## Données

L'ensemble de données se trouve dans le répertoire **src/assets/data/** dans l'archive fournie pour le travail pratique. L'ensemble de données contient les colonnes suivantes :

- **Arrond**: ID du quartier où l'arbre a été planté.
- **Arrond\_Nom**: Le nom du quartier où l'arbre a été planté.
- **Date\_Plantation**: La date à laquelle l'arbre a été planté.
- **Longitude**: La longitude de l'arbre.
- **Latitude**: La latitude de l'arbre.

## Prétraitement des données

Pour commencer, vous devrez traiter les données que nous vous fournissons. Les données contenu dans le fichier CSV sont brutes, il est donc nécessaire de réorganiser certaines parties de celui-ci afin qu'il puisse être correctement utilisé par la bibliothèque D3. Pour ce faire, vous devez compléter le fichier [scripts/preprocessing.js](#).

Plus précisément, vous devrez effectuer ces étapes:

1. Lisez les données et extrayez une liste de noms de quartiers (fonction [getNeighborhoodNames](#))
2. Filtrez les données par année (fonction [filterYears](#)). Bien que nous ayons des données sur les arbres plantés à partir du début du 20e siècle, nous voulons nous concentrer sur les plantations d'arbres entre 2010 et 2020 pour cette visualisation.
3. Regroupez et restructurez les données pour obtenir le nombre total d'arbres plantés par an et par quartier (fonction [summaryYearlyCounts](#))
4. Remplissez les données manquantes avec la valeur 0 (fonction [fillMissingData](#))

L'extrait 1 ci-dessous montre une portion de la table de données qui devrait résulter de ces étapes. L'extrait montre l'un des objets qui devrait être dans le tableau.

```
{  
  "Arrond_Nom": "Ville-Marie",  
  "Plantation_Year": 2017,  
  "Comptes": 888  
}
```

*Extrait 1: Objet résultant pour le quartier Ville-Marie en 2007*

## Carte de chaleur

Pour cette deuxième partie, vous devrez implémenter la partie principale de la visualisation de données. Tout d'abord, vous devrez mettre à jour l'échelle de couleurs utilisée pour colorier les rectangles dans la visualisation. Deuxièmement, vous utiliserez les données pour ajouter les rectangles à la carte de chaleur. Troisièmement, vous allez mettre à jour les échelles x et y en fonction des données et de la taille de visualisation et dessiner les axes en conséquence. Quatrièmement, vous mettrez à jour la position, la taille et la couleur des rectangles que vous avez précédemment ajoutés, ce qui aboutira à la carte de chaleur terminée. Pour atteindre cet objectif, vous devez compléter le fichier [scripts/viz.js](#).

Voici les étapes que vous devrez accomplir pour cette partie:

1. Mettez à jour le domaine de l'échelle de couleurs pour refléter les données (fonction [setColorScaleDomain](#))
2. Ajoutez la structure SVG et liez-y les données (fonction [appendRects](#)). Cette structure sera utilisée pour dessiner la carte de chaleur dans les étapes suivantes.
3. Mettez à jour les échelles utilisées pour déterminer les positions en x et en y pour refléter les données et la taille de la carte de chaleur (fonctions [updateXScale](#) et [updateYScale](#)). Assurez-vous que les noms des quartiers sont classés par ordre alphabétique.

4. Dessinez les axes x et y respectivement en haut et à droite du graphique, avec les graduations sur l'axe x pivotées de 45 degrés (fonctions `drawXAxis`, `drawYAxis` et `rotateXTicks`)
5. Mettez à jour l'apparence visuelle des rectangles, en les positionnant et en les dimensionnant selon les échelles x et y (fonction `updateRects`). Leur couleur doit être choisie en fonction de l'échelle de couleurs.

La figure 1 illustre l'apparence attendue de la carte de chaleur après ces étapes.

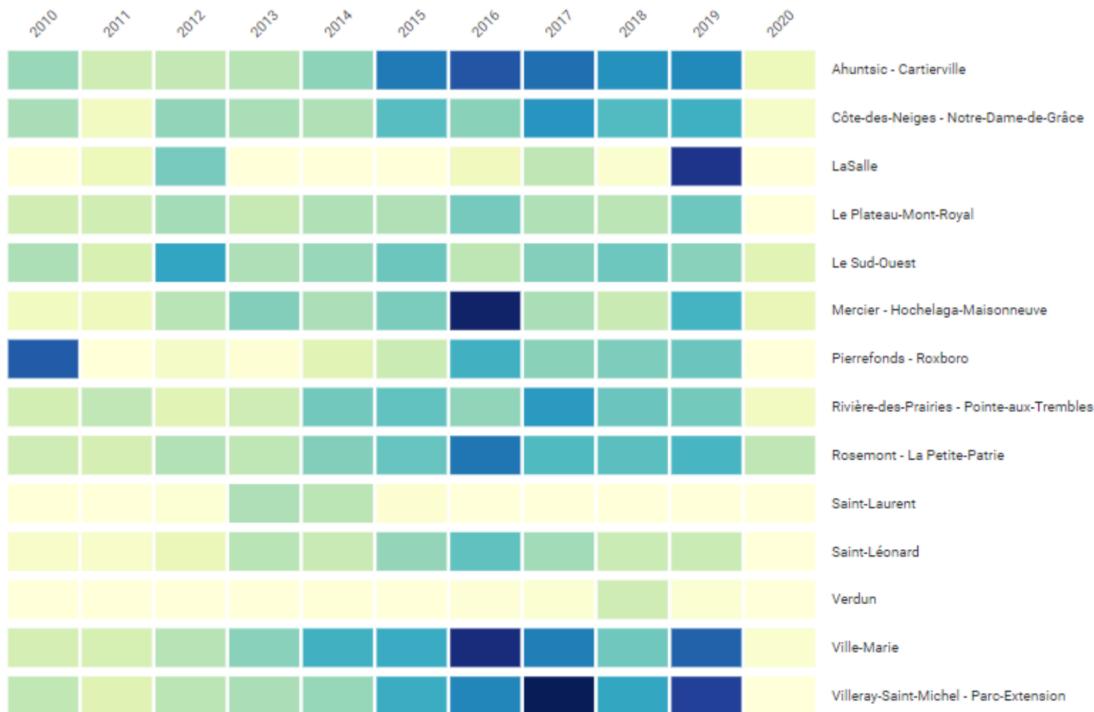


Figure 1: La carte de chaleur

## Légende

Pour cette troisième partie, vous allez générer une légende verticale à gauche de la heatmap. La légende indiquera le nombre d'arbres plantés pour chaque couleur dans l'échelle de couleur continue. Une partie du code de cette partie a déjà été implémentée.

Le code fourni initialise le dégradé utilisé comme remplissage de la légende, initialise le rectangle pour représenter la barre de couleur de la légende et initialise le groupe pour l'axe de la légende. Votre tâche consistera à dessiner la légende elle-même à l'aide de ces éléments. Pour remplir cette partie, vous devez compléter le code dans le fichier `scripts/legend.js`.

L'étape à suivre pour accomplir cette partie est la suivante:

1. Dessinez la barre de couleur de la légende avec son axe correspondant (fonction `draw`)

Le résultat de cette étape est illustré à la figure 2. Portez attention à l'apparence visuelle de chaque élément représenté.

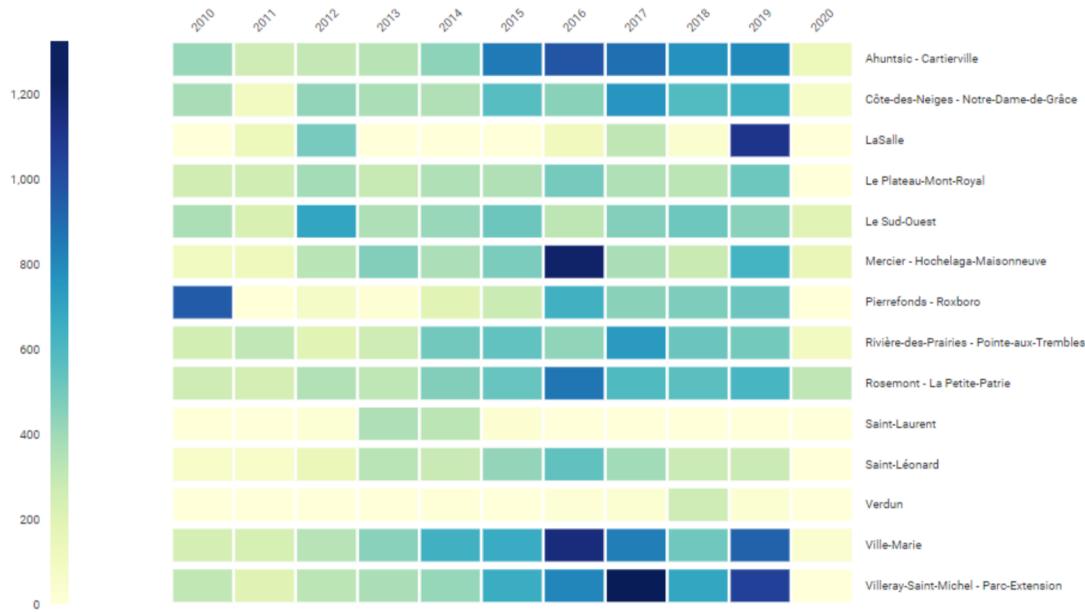


Figure 2: La carte de chaleur avec la légende qui l'accompagne

## Survol

Pour cette quatrième partie, vous allez implémenter le comportement de survol de la carte de chaleur. Le but de cette fonctionnalité est de faciliter la lecture de la carte de chaleur. Lorsqu'un rectangle est survolé par la souris, nous souhaitons que son compte d'arbres correspondant soit affichée en son centre. Les graduations sur les axes x et y qui correspondent au rectangle survolé doivent également afficher leur texte en gras tant et aussi longtemps que le rectangle reste survolé par la souris. Le code à compléter pour cette partie se trouve dans le fichier [scripts/hover.js](#).

Plus précisément, les étapes que vous devrez effectuer pour cette partie sont:

1. Appelez les fonctions appropriées lorsqu'un rectangle est survolé par la souris(fonction [setRectHandler](#))
2. Lorsqu'un rectangle est survolé, affichez son nombre d'arbres en son centre (en noir, ou en blanc si la valeur est de 1000 ou plus). Assurez-vous que le rectangle est affiché avec une opacité de 75% en survol et une opacité de 100% sinon (fonctions [rectSelected](#) et [rectUnselected](#))
3. Pendant qu'un rectangle est survolé, mettez les graduations correspondantes sur les axes x et y en gras (fonctions [selectTicks](#) et [unselectTicks](#))

Les figures 3 et 4 qui suivent illustrent le comportement attendu lorsqu'un rectangle est survolé par la souris.



Figure 3: La carte de chaleur avec le quartier Rivière-des-Prairies - Pointe-aux-Trembles et l'année 2015 survolés

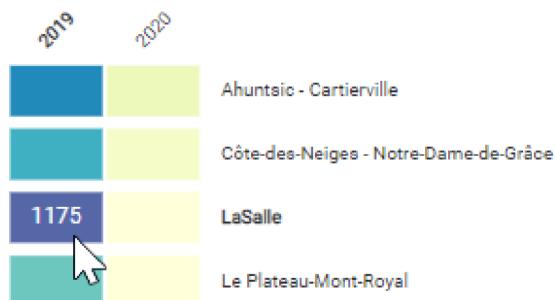


Figure 4: La carte de chaleur avec le quartier LaSalle et l'année 2019 survolés

## Soumission

---

Les instructions pour la soumission sont:

1. Vous devez placer le code de votre projet dans un fichier ZIP compressé nommé [matricule1\_matricule2\_matricule3.zip]
2. Le travail pratique doit être soumis avant le [29 Mai 23h59]

## Évaluation

---

Dans l'ensemble, votre travail sera évalué selon la grille suivante. Chaque section sera évaluée sur l'exactitude et la qualité du travail.

| <b>Exigence</b>                            | <b>Points</b> |
|--|---------------|
| Prétraitement des données                  | 4             |
| Carte de chaleur                           | 8             |
| Légende                                    | 3             |
| Survol                                     | 4             |
| Qualité globale et clarté de la soumission | 1             |
| <b>Total</b>                               | <b>20</b>     |

## Références

---

[1] Service des grands parcs, du Mont-Royal et des sports, "arbres sur le territoire de la Ville ,," Montréal : Portail de données ouvertes. Available: <http://donnees.ville.montreal.qc.ca/dataset/arbres> [Accessed 01 09 2020].